King Saud University
College of computer and Information Sciences
CSC 311 – Design and Analysis of Algorithms

جامعة الملك سعود
كلية علوم الحاسب والمعلومات
311 عال –

Midterm I Exam - Fall 2019 | Wednesday October 9th, 2019 | Exam time: 05:00-6:30 P.M.

Student's name: عبدالرحمن علي العطا... ID: [redacted] Section: 10-11

$17.5/20$

Problem 1 (3.5 points) $3.5/3.5$

(a) Give the following functions a number in order of increasing asymptotic growth rate. If two functions have the same asymptotic growth rate, give them the same number.

	Function	Rank
$n^2 <$	$\frac{n^2}{\lg n}$	2
$\log n$	$5 \lg n^8$	1
n^2	$\lg n^4 + 5n^2 - 100$	3
n^3	$8^{\lg n} + 2n^2$	4
$\log^2 n$	$2 \lg n + \lg(\lg n^2)$	1
2^n	2^n	5

(b) Using the definition of θ, find $g(n)$, C_1, C_2, and n_0 in the following:

$$5n^4 - 2n^3 + n^2 \in \theta(g(n))$$

$5n^4 - 2n^3 + n \le 8n^4$

$n_0 = \frac{3}{5-2} = \frac{3}{3} = 3$

$\boxed{C_1 = 8 \quad C_2 = 2 \quad n_0 = 1}$

$g(n) = O(n^4)$

...em 2 (5 points) 4/5

a- What is the time complexity of the following algorithm? Prove your answer.

```
...thod1 (arr[], low, high)

  pivot = arr[high];                                       1
  i = (low - 1)  // Index of smaller element               1
  for (j = low; j <= high- 1; j++)                         n+1
  {
      if (arr[j] < pivot)                                  n
      {
          i++;    // increment index of smaller element    n
          swap arr[i] and arr[j]                           n
      }
  }
  swap arr[i + 1] and arr[high]                            1
  return (i + 1)                                           1
```

2	5	1	3	4

$i=1$

1	2	3	4	5

$T(n) = 4n+5$

$4n+5 \leq 9n$

$c_1=9 \quad n_0=1$

$T(n)=O(n)$

b- What is the recursive equation corresponding to the following pseudo-code. ?

```
MySol(arr[], low, high)
              1     n
{
    if (low < high)     1
    {
        pi = Method1 (arr, low, high);  1
        MySol (arr, low, pi - 1);  T(n/2)
        MySol (arr, pi + 1, high);  T(n/2)
    }
}
```

$T(n) = 2T(n/2) + \cancel{1}$ - 1

c- Prove the big-Oh performance of the MySol.

$T(n) = 2T(n/2) + 1$

$= 2[2T(n/4) + 1] + 1 \qquad = 4T(n/4) + 2 + 1$

$= 4[2T(n/8) + 1] + 2 + 1 \qquad = 8T(n/8) + 4 + 2 + 1$

stop when $\frac{n}{2^k} = 1$

$n = 2^k$

$= 2^k \cdot T(1) + \sum_{i=0}^{k-1} 2^i$

$= 2^k + \frac{1 - 2^k}{1 - 2}$

$n + n - 1$

$= 2n - 1$

$T(n) = O(n)$

7

Problem 3 (2 points) 2/2

Consider the following recurrence relation. Solve it using recursive substitutions and find its asymptotic performance.

$$\boxed{\frac{1-\log n}{-1}}$$

(a)

$$T(n) = 2T(\tfrac{n}{3}) + n^2.$$

تكملة

$T(n) = 2T(n/3) + n^2$

$= 2[2T(n/9) + \frac{n^2}{3}] + n^2$

$= 4T(n/9) + \frac{2n^2}{3} + n^2$

$= 4[2T(n/27) + \frac{n}{9}] + \frac{2n^2}{3} + n^2$

$= 8T(n/27) + \frac{4n^2}{9} + \frac{2n^2}{3} + n^2$

Stop when $\frac{n}{3^k} = 1$

$n = 3^k$ / $\log n = k$

$2^k T(1) + n^2 \sum_{i=0}^{k-1} \frac{2^i}{3^i}$

$n + n^2 \left[\frac{\frac{1-2^k}{1-2}}{\frac{1-3^k}{1-3}} \right]$

$= n + n^2 \left[\frac{2(n-1)}{n-1} \right]$

$= n + 2n^2 \leq 3n^2$

$c = 3$

$n = 1$

$T(n) = O(n^2)$

Problem 4 (1.5 points) 1/1.5

Solve the following recurrences using the recursion tree method. Justify your answers.

$$T(n) = T(n/2) + 3$$

$(2^{k} - k^{+1})3$

$T(n) = T(n/2) + 3$ ✓ $k=1$

$= 2[T(n/4) + 3] + 3$ ✓

$k=2$ $\boxed{3+2\times3+2\times3}$

$= 2T(n/4) + 3\times3$

$= 2[T(n/8) + 3] + 3\times3$

$= 2T(n/8) + \cancel{5\times3}$ $k=3$ -0.5

Stop when $\frac{n}{2^k} = 1$

$n = 2^k$

$\log n = k$

$= 2T(1) + 3 + 3\sum_{i=1}^{k-2} 2$

$= 2 + 3 + 3(2(k-1))$

$= 2k - 1$

$= 2\log n - 1$

$T(n) = O(\log n)$

...tem 5 (5 points)

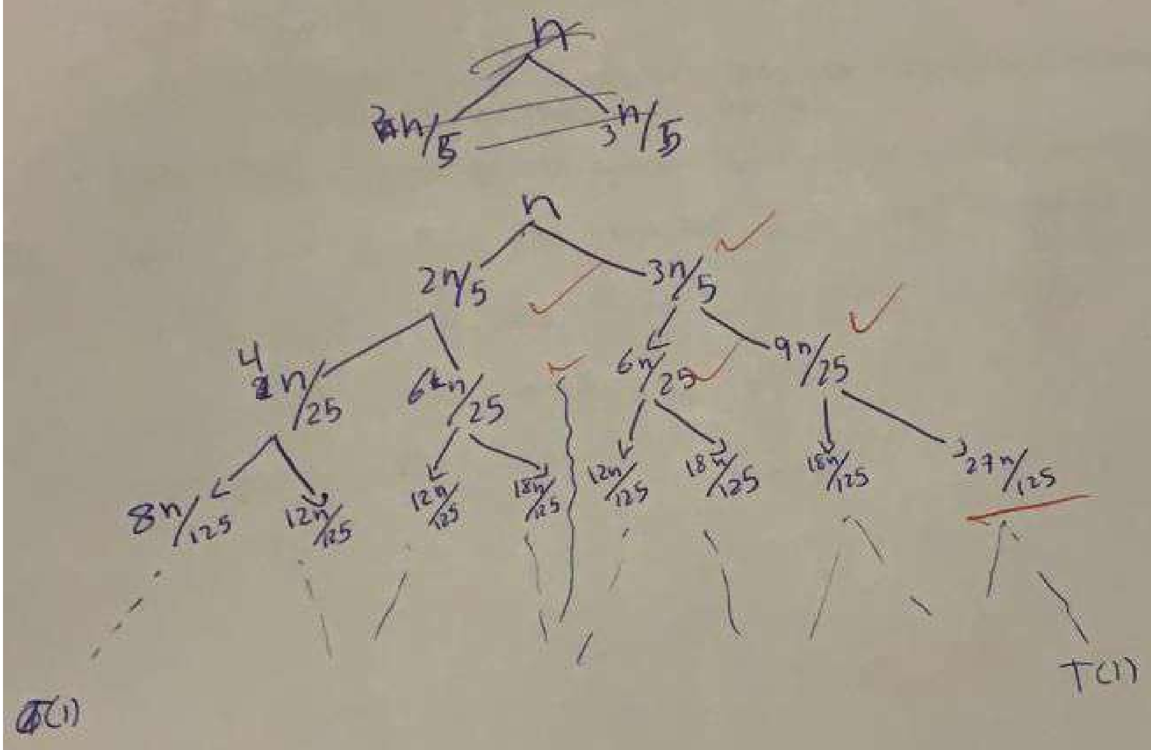
5/5

Consider a variation of *MergeSort* which divides the list of elements into two lists of size **2/5** and **3/5**, recursively at each step, instead of dividing it into halves. The Merge procedure does not change.

(a) Give a recurrence relation for this algorithm

$$T(n) = T\left(\frac{2n}{5}\right) + T\left(\frac{3n}{5}\right) + n$$

(b) Draw a recursion tree for the algorithm



stop when $\frac{n}{5^k} = 1$

c) Using the recursion tree, explain how you can deduce the worst case upper bound.

the sum of all row $\leq n$ and ~~the~~ the length of the tree is $\log_{5/3} n$, the last element is $T(1)$ ~~with~~ with length $= \log_{5/3} n$ and the sum of that row is n

sum x length $= O(n \log_{5/3} n)$
⇓
worst case

Problem 6 (3 points) 2/3

Consider the following problem:

There are ***n*** parking spots numbered from 1 to ***n*** and you are told that there are ***k*** cars in the first ***k*** spots (one car in each spot), and all other spots are empty. How to find the value of ***k***?

a- Suggest a **divide and conquer** algorithm and give its high-level description to solve the problem (find ***k***)?

we will divided the number of parking to ~~substitute~~ subarrays and check if the $(1+n)/2$ middle is empty or not and do it untill find the value of K then we will conquer it.

pseudo Code ?